Raise The Bar:
Video-Based Barbell Path and Velocity Tracking Application for Weightlifting

Neython Lec Streitz
nlecstreitz@oxy.edu
Occidental College

1 Problem Context

From the average gym-goer to the elite professional athlete, the ability to track and monitor training is key for supporting safe and effective workouts. Specifically for intermediate to advanced athletes, exercise efficiency and load management becomes crucial for maintaining long-term performance improvements. The purpose of this project is to provide an accurate, useful, and accessible method of tracking the path and velocity of a barbell during resistance training exercise via video-recorded exercises. Through the use of this application, athletes and coaches are able to objectively track the range of motion and intensity of any barbell squat exercises they perform.

Put simply, bar path refers to the range of motion of the barbell during an exercise. Bar path tracking thus, tracks the time and distance of the barbell during a specific lift. By optimizing the bar path for a given exercise, an athlete is able to pinpoint weak points in their movement and generate better efficiency overall. Especially for beginners, bar path is an incredibly helpful tool for improving and visualizing exercise form.

For the more advanced, velocity tracking provides even more granularity. Typically, exercise loads are prescribed using percentage-based methods. For example, a coach might make their athlete perform 5 sets of 5 repetitions using 70% of their 1-rep max. Percentage-based training has a long history of being an effective approach for resistance training design. Recently, Velocity Based Training (VBT) has emerged as a contemporary method of resistance training design, using the velocity of the barbell during an exercise to supplement and inform training decisions.

VBT is useful because it allows for more athlete-specific load (weight amount) and volume (repetition amount) prescriptions [28]. For one, it can gauge an athlete's current fatigue levels by comparing velocity in previous sessions. This practice is known as autoregulation, where an athlete can pull back on high fatigue days and push harder on days they are feeling fresh. Secondly, VBT allows coaches (or self-lead gym-goers) to assign training loads that deliver more precise and objective levels of effort and fatigue. Some athletes have difficulty doing higher repetition work, whereas others see sharp drop-offs near their 1RM. In the percentage-based approach, set and repetition recommendations do not account for this variability [1]. With all that in mind, bar path and velocity metrics serve as a way to standardize technique for a beginner, and for the advanced, give biofeedback in real-time and enable the design of athlete-specific programs.

Currently, there exists various hardware and software-based approaches to tracking bar path and velocity. This includes hardware devices like linear position transducers and 3D motion analysis systems, as well as video-based smartphone apps. However, using hardware to track velocity is expensive and inaccessible for the majority of regular gym-goers and even for advanced athletes. Furthermore, most smartphone apps show poor accuracy and clunky design [18, 14]. Thus, our project is in a position to improve on the accuracy and design of existing smartphone apps, while circumventing the money and infrastructure needed for hardware-based approaches.

2 Technical Background

At its most basic, the question at hand is how we can track, and ultimately display, the position and rate of change of a dynamic object in real time. Thus, this project is one in the realm of computer vision (CV), or, how computers interpret the visual world. At a glance, the technical components of this project include object detection and object motion tracking.

2.1 Object Detection

To begin, there needs to be a way to detect the barbell within the frame of the video. The main goal of object detection within CV tasks is to identify and draw boundaries around objects in a given image or video. State-of-the-art approaches utilize deep-learning methods and neural networks to perform object detection. To this end, there are typically two main parts to object detection with deep-learning. Firstly, the image or frame passes through an encoder that extracts statistical features used to locate and label objects. Then, the encoder's outputs are passed through a decoder that predicts the boundaries and labels of each of the objects.

An existing network architecture is usually the encoder used for object detection. Two of the most common network architectures are VGG and ResNet [24, 10]. However, these networks are simply too large for resource constrained devices like smartphones. Developed by Google, MobileNet is an encoder specifically made to work on mobile devices [12]. It functions by employing depthwise separable convolution, a process that decreases the amount of parameters in the network, making it less resource intensive.

Common decoders used for object detection applications include Faster R-CNNs and You Only Look Once (YOLO) [22, 21]. While Faster R-CNNs delivers great accuracy, it suffers in overwhelming complexity and slow speed. YOLO has the opposite problem, as it provides speed but lacks in accuracy. Google's Single Shot Detectors (SSD) aims to provide a middle ground [17]. SSD works by laying a grid of anchor points over the input image then overlaying boxes of multiple shapes and sizes at each anchor point. The model then predicts the likelihood that the object to be detected exists within the box, changing the boxes to fit the object as best as possible. The lower likelihood boxes are pruned away, leaving a best prediction of the boundaries of the object.

By combining MobileNet and SSD, smartphones are capable of supporting deep learning-based object detection. Having said that, the MobileNet and SSD combination supports the detection of 20 different objects, none of which are barbells. Given that the process of training a network to specifically detect barbells is likely very GPU intensive, slow, and unreliable, this project takes advantage of the relative ease of tracking pre-specified markers. Specifically, we take advantage of fiducial marker detection techniques. These markers are synthetic square markers composed of a black border with an inner binary codification. AruCo tags, a form of fiducial marker, are incredibly great for estimating camera pose. The AruCo library was first developed by Rafael Muñoz-Salinas and Sergio Garrido-Jurado [8]. The square shape of AruCo tags provide the necessary correspondences to estimate the position and orientation of the barbell. Furthermore, because of their inner codification, AruCo tags add extra reliability in detection. Adding fiducial markers to the end of barbells, essentially turns the problem of barbell detection into one of contour detection.

A contour is a well-defined boundary with typically matching color or intensity within. In the case of fiducial markers, there is a bit pattern within the outline. To allow for contour detection, the image or video frame is first converted to grayscale. Next, binary thresholding is applied, converting the image to black and white by setting only the borders of the object white and the rest black. A contour algorithm is finally applied to identify the borders, and if needed, draw them out over the original image. The algorithm simply detects potential shapes that match the outline of the contours. Once detected, the contours are analyzed by extracting the marker bits within the marker. A perspective transform is applied to isolate the marker. The image is further divided into separate cells and bits within each cell are counted. The bit counts are matched against a dictionary to determine validity as a marker. Contour detection is thus a much simpler process, allowing for ease-of-use while preserving accessibility due to not needing to train a novel neural network.

2.2 Object Tracking

In essence, the velocity of a barbell is simply its directional movement over time. To get the barbell velocity, it is necessary to track it throughout the frames of a video. In the case of a barbell, we will be engaging in Single Object Tracking (SOT).

Typical problems in object tracking include occlusion, detection speed, and spatial scale variation. Occlusion is simply when objects are so close to each other that they appear to be one merged object. Luckily, supporting occlusion is not something necessary to deal with, as the barbell will almost always be in an open and unimpeded space. Similarly, because the barbell is tracked from a side-on view, almost as if it was two-dimensional, there is little variation in object scale. Detection speed is certainly an issue, but because this is a contour detection problem, there already exists fast and efficient techniques.

There are a variety of neural network approaches to object tracking, including BOOSTING Tracker, MIL Tracker, and KCF Tracker [9, 4, 11]. These approaches each have their benefits and downsides, typically balancing general accuracy, occlusion support, lighting and viewpoint change, and error detection.

For the purposes of this project, tracking will involve repetitively updating the camera and calculating the previous pixel position with the current. Knowing the perimeter measurement of the AruCo tag in real life, finding the calculated digital perimeter of the tag in each frame can be used to convert the pixel movement per frame of the barbell to its real life m/s equivalent. A list-like data structure such as the *deque*, provides quick enough pops and appends to maintain and retrieve the previous positions of the barbell. Due to the frame rate of modern smartphones, this process should be quick enough to somewhat accurately track the position and velocity of the barbell.

3 Prior Work

As mentioned before, there currently exists a variety of hardware and software-based methods for tracking barbell path and velocity. While the market is fairly large, there are only a handful of devices, systems, and apps that have been studied in sports science literature. The studies that have been done, mainly evaluate the reliability, ease-of-use, and practicality of these technologies.

Before discussing explicit barbell path and velocity tracking projects, it is important to mention works related simply to fiducial marker detection. Prime among these include work out of the University of Michigan by Wang et al., where they developed an improved algorithm for AprilTag detection. AprilTags are specialized fiducial markers used in many robotics applications because of their utility in camera pose estimation. They are black and white square tags with an encoded binary pattern. The improved algorithm takes advantage of the rarity of occluded tag detection, allowing the algorithm to function quicker. Furthermore, Wang et al., showed detection improvements at a wide variety of distances away from the tags. The AprilTag system is based on even earlier work including ARTag and ARToolkit [15, 7]. However, AprilTags are less straightforward to set up than the chosen AruCo tags, due to having no implementation with the OpenCV python library. Furthermore, more parameters in the AprilTags means higher false detection rates. Ultimately, these fiducial tag systems and detection algorithms are efficient, applicable in a wide variety of CV problems, and maintain high accuracy and resilience against false positives.

As far as hardware-based approaches are concerned, linear position transducers, also known as linear encoders, are considered the gold standard for barbell velocity tracking. These devices clip onto the barbell and have a cable moving in and out of the device. Examples of linear encoders available on the market include Vitruve, T-Force, and GymAware. These companies also include or offer paid mobile apps for synchronization with their devices. All three of the listed devices have been validated by studies in sports science literature [18, 26, 19]. However, great accuracy comes at a price, as linear encoders are expensive and inaccessible. The lone GymAware device comes in at over $1,995, with accompanying software starting at an additional $495 minimum. Vitruve is a cheaper option, yet still comes in at around $397 for the device and mobile app. One more thing to note is the potential for damaging the devices because they must remain directly under the barbell. This is especially problematic for intensely dynamic movements like the Olympic lifts.

There are currently no hardware-based devices on the market with the express purpose of visualizing barbell paths. That's not to say that three-dimensional motion capture camera systems cannot and have not been used to track bar path and velocity [19]. Yet, like linear encoders, camera systems are costly and involve infrastructure not accessible nor feasible to set up for the average gym-goer. Nonetheless, the reliability of these devices means they are often used to test the validity of novel velocity trackers, like mobile apps [19].

For software-based approaches, specifically, mobile applications, there are multiple offerings on the market. Examples of apps that include both bar path and velocity tracking include MyLift, IronPath, and BarSense. Of those, MyLift and IronPath have been studied in the literature. Markedly, studies have concluded that both MyLift and IronPath are not reliable tools for the measurement of barbell velocity. Kasovic et al. showed that the IronPath app compared with a linear transducer, "recorded significantly lower average concentric velocity values for the front squat and back squat and greater ROM values for the sumo deadlift" [14]. Similarly, Martinez-Cava et al. concluded that "the excessive errors of the newly updated MyLift app advise against use of this tool for velocity-based resistance training" [18]. With that said, other apps as well as experimental software approaches have shown promising results.

In a paper by Pueo et al., they developed a smartphone-based camera system for automatically detecting and tracking a barbell using a custom image-processing algorithm [20]. The algorithm works by creating reference points on the parallel supports of the exercise machine, in this case, a Smith Machine, to segment and compute the barbell position without prior calibration. Their system showed near matching performance with a linear position transducer. That being said, their paper also admitted that the video system is limited by the use of a barbell machine, and does not support free weight barbell movements. Moreover, their system does not include any path tracking and visualization features. Nonetheless, in terms of velocity tracking accuracy, their paper is certainly something to look up to.

Interestingly, two papers by Balsalobre-Fernández et al. concluded that the MyLift (formerly Powerlift) app was in fact a valid and reliable method of tracking barbell velocity [2, 3]. The two studies showed high correlation between MyLift and a linear position transducer. However, a study and subsequent response by Courel-Ibañez et al. contradicted these findings, showing poor reliability from the MyLift app [5]. They also noted that the main author of the two papers where MyLift was validated, is also the main developer of the app. Combined with the aforementioned study by Martinez-Cava et al., that also advised against its use, the MyLift app does not serve as an example of a validated smartphone-based velocity tracker [18].

Ultimately, while linear position transducers deliver incredible accuracy and precision, they are far too expensive to be accessible tools for regular gym-goers and most athletes to use in the gym. Not to mention, linear position transducers also lack in the visualization of barbell path and have the danger of being damaged by the barbell. Unfortunately, due to their consistency, very little work is being done to improve on their flaws. Otherwise, several mobile apps already exist on the market that offer both barbell path and barbell velocity tracking for free. Yet, their inability to track velocity with accuracy and precision makes them unreliable for the practice of velocity based training methods. Pushing this field of technology forward means developing apps with sufficient precision to implement VBT, while maintaining ease-of-use, quality design, and accessibility to a wide audience of gym enthusiasts and athletes.

4 Methods

As a problem of computer vision, the majority of the technical aspects of the project have to do with identifying and then tracking the barbell. Additionally, automating the detection of repetitions was prioritized in the structure of the code. As mentioned before, we have chosen to use AruCo tags which are easily printable and placed on a wide variety of barbells. This is cheap, removes the need to color and/or permanently mark a barbell (that might be for public use), and has been used for other CV applications with great success. Velocity and bar path tracking involves the constant identification and/or optical flow estimation of the bar over time. Each technical aspect, barbell identification, bar path tracking, and automatic rep detection is detailed below.

4.1 Barbell Identification

Inspired in part by a project out of the University of Michigan's April Robotics Lab, our barbell identification relies on fiducial marker identification [27]. Why AruCo tags specifically? As low cost is a primary goal of the project, by using AruCo tags we can remove the need for training novel machine-learning-based barbell identification models while still maintaining increased accuracy. AruCo tags are easy to generate, cheap and printable, and more efficient to stick/remove from barbells than permanent options (i.e, coloring the barbell). Three other contour techniques were considered but ultimately discarded: colored contours, QR codes, and AprilTags. While brightly colored stickers are incredibly motion blur resistant and quite easy to detect, AruCo tags provide stronger error detection within a busy and often colorful gym setting. Additionally, QR tags were considered but were found to be more difficult to detect than AruCo tags and more susceptible to motion blur. Lastly, the original inspiration for the method, AprilTags, were considered but ultimately not followed due to being much more difficult to set up than AruCo tags. Thus, AruCo tags were chosen as the best candidate for barbell identification.

With the AruCo tags in mind, the first step is identifying the barbell. To successfully tape the AruCo tag to the barbell, a small piece of cardboard is used. It is important to ensure that the AruCo tag has sufficient white space to contrast its wide black border. A more advanced version also includes a small section of a paper towel roll taped to the back of the cardboard to allow the tag to be taken on and off of the barbell at will. For a user, the project has a feature for generating tags to print. To note is that users will still have to construct the cardboard holder themselves. The AruCo tag will then be taped centrally onto the end of the barbell facing the camera.

After starting the video recording, the AruCo tag detection follows with two main steps. Firstly, the image is processed via grey-scaling and adaptive thresholding, segmenting the current frame by pixel intensity. Contours are then extracted from the thresholded image, with special attention paid to only those contours that match the general outline of the marker. This processing essentially encompasses marker candidate detection. Secondly, to determine whether or not the candidates are truly markers, their inner codification is verified. The bits of each marker are extracted, counted (white and black bits), and then checked against a dictionary to determine validity. Fortunately, the barbell is recorded directly parallel, meaning no extra work needs to be done to determine orientation or solve for occlusion.

While the default settings of the AruCo detection algorithm was used, testing was done to determine the best fit from a AruCo dictionary perspective. Originally, the 4x4, 50 marker library was used to generate and detect tags. There are multiple grid sizes as well as total marker dictionaries to choose from. Choosing less total markers allows for greater differences marker-to-marker, however since only one marker exists within the frame, choosing 50 total markers versus 250 total markers makes a trivial difference. The 4x4, 50 marker dictionary, means any AruCo tag generated contains a 4x4 inner codification and comes from a selection of 50 possible tags. Through testing, this dictionary did not meet the error detection necessary for our application. Other square objects within frames were detected as tags causing the algorithm to calculate skewed coordinate positions and as a result, skewed velocity outputs. On the opposite end, the 8x8, 50 marker dictionary was too susceptible to motion blur due to its heavy granularity relative to other markers. This was expected considering the previous difficulty with QR tags. As for the final dictionary, we found the 6x6, 50 marker library to provide the best balance of ease-of-detection and error robustness.

4.2 Bar Velocity and Path Tracking

By repeatedly identifying and locating the AruCo tag frame-by-frame, we are able to track the barbell over time, giving us the necessary path and velocity output. The algorithm begins by initializing, waiting for a tag to appear in the frame. Since the camera should already be set up to view the barbell end, this process will happen rather quickly. Next, we perform AruCo tag detection as outlined in the above subsection, saving the detected bounding corners. Even with a simple tag, motion blur can still be an issue during very fast movement. Thus, optical flow estimation is used to fill-in any gaps in detection. Optical flow estimation works by performing feature matching and determining where the previously detected marker corners have moved in the next frame. That is, it identifies the differences in the original frame to next frames, and estimates the pixel movement between the frames. Given that the real perimeter of the tag is known and each frame is 1/fps seconds in length, the pixel distance frame to frame along with the pixel perimeter of the AruCo tag is enough to calculate mm/s every 1/fps. From there, we calculate the average concentric velocity, peak velocity, displacement, and velocity loss. This process is repeated continually until the user ends the video.

The contrail of the barbell (the bar path) is relatively straightforward to implement. During the bar localization process, we save the past and current points to compute the displacement and draw lines connecting the points. Velocity tracking similarly takes the past and current location points to determine distance over frames.

4.3 Automatic Rep Detection

One of the biggest issues with existing barbell velocity and bar path tracking applications is the need for heavy manual interaction. This includes but is not limited to identifying the barbell at different points in the video, tracing the bar path manually, and/or specifying the amount of reps performed. To combat this, an automatic repetition counter was implemented using the coordinate data computed by the previous methods. Two specific features are necessary to apply the automation. Firstly, its important to note that most lifters will set up their video before engaging in the set. This means that they will likely need to unrack the barbell before performing the exercise. Movements such as racking or unracking have the potential to confuse the rep detection algorithm. Clearly, greater horizontal movement than vertical movement indicates racking/unracking of the barbell. Thus, the positional history and velocity history is erased if this horizontal bias is detected.

Next, rep detection was implemented by splitting the two phases of a lift and detecting them within the positional data. The eccentric portion, as the lifter lowers the barbell is indicated by an inflection downwards. The concentric portion, as the lifter raises the barbell is indicated by an inflection upwards. Detection of inflection points at the top of the rep and at the bottom signify the completion of a single repetition, and aggregate statistics are calculated after each inflection pair is noticed. Unfortunately, this setup means other types of lifts are not currently possible with the app. This happens for two main reasons. Firstly, exercises like the bench press for instance, do not have the same magnitude of movement that back squats do so the algorithm is poor in detecting unracking/racking. Secondly, exercises like the deadlift move in the opposite direction of a squat, starting with the concentric and ending with the eccentric. Nonetheless, for the barbell back squat, automatic rep detection works for all reasonable sets of the exercise.

5 Evaluation Metrics

For the purposes of this project, only velocity accuracy was tested via a small study. To determine the accuracy of velocity tracking, a study was conducted on a small sample of participants comparing the app's output with an existing hardware device on the market. Due to this focus, no app-focused user testing was performed. Instead, participants in the accuracy study were asked to share thoughts on possible features after completion of their sets. Below, we delineate participant selection, study design, and study methodology. Results from the study and feedback from participants is discussed in the following Results section. The full dataset of velocity comparisons is available to view within this paper's repository.

5.1 Participant Selection

We selected five users to participate in the evaluation of velocity tracking accuracy. Users were made aware of the training design and procedure. Users were also informed of the basic advantages of using velocity-based training methods. No participant reported any injuries or limitations prior to beginning the study. In total, three participants identified as men and two as women. Of the three men, all considered themselves intermediate-advanced lifters with 3+ years training. Of the two women, both considered themselves to be novice lifters with scattered/inconsistent training histories. Lastly, all participants were familiarized with the barbell back squat exercise used in the study.

5.2 Study Design

Participants utilized the app for just the barbell back squat. Prior to starting, participants were educated on the benefits of velocity-based training as well as some common limitations to the practice. After finishing the introduction, participants began with the active portion. For the exercise, participants performed a progressive loading test consisting of increased loads of 20%, 40%, 60%, and 80% of their estimated 1RM. The reason for the gradual increase in barbell weight was to ensure reliability is covered for a variety of velocity ranges. Similar, but more precise, work was done by Martinez-Cava et al. [18].

After each set, participants took 5 minutes of rest to recover and analyze their bar path and velocity metrics outputted from the app. Results in average concentric velocity and peak velocity were saved to be analyzed. Building off work by Tomasevicz et al., we calculated simple Pearson correlations between our app and the Push Band 2.0, now owned by Whoop [25].

5.3 Study Methodology

After a familiarization session with the exercises and barbell, participants were shown the app with a sample prerecorded video. Through this showing, participants were informed of the benefits and limitations of velocity-based training and velocity tracking in general. Participants were asked previously to give their one-repetition maximum to ensure all participants could complete the study with each given weight.

Next, participants were asked to perform five repetitions at maximum velocity for each weighted increase with 5 minutes of rest in between. An iPhone smartphone was used to record each set, and it was kept in the same location at the same angle to the side of the squat rack for all participants. This recording was done at 60 fps. In between sets, participants were allowed to view the set within the app on a separate laptop. This ensured that the smartphone was not moved to a different position in between sets or between participants. During this time, the velocity statistics were recorded from the Push Band 2.0 and after the participant was done, from the smartphone as well (after being uploaded to the app).

6 Results and Discussion

The stated goals of the project are to deliver a cheap, accessible, and accurate velocity and path tracking application for barbell-based exercise. On the whole, AruCo tags are quite inexpensive to print and only require some form of adhesive to place on a barbell. Thus, the resources necessary to use the velocity tracking are far less expensive and far more accessible than a linear position transducer. However, our project is slightly more expensive and slightly less accessible than smartphone apps on the market, given that a tag must be printed, whereas those apps require nothing but a phone. Nonetheless, this added cost should come with a much needed increase in accuracy compared to the smartphone apps on the market. Furthermore, it allows for a more automatic detection of the barbell and of squat repetitions.

6.1 Velocity Accuracy

To evaluate the accuracy of the app, we will follow standard convention and compare it to an already validated device, the Push Band 2.0. As stated by Janicijevic et al., LPT's, specifically GymAware and T-Force have been validated multiple times in the literature "as gold-standards in studies designed to validate other velocity monitoring devices"[13]. However, without access to LPT's, the Push Band was the next best option. It is important to note that studies have not completely validated the Push Band device. A study by Balsalobre-Fernández et al., showed that while the Push band was accurate enough to have practical applications in velocity based training, it was unsuitable for very light loads and at faster velocities [2]. Thus, any accuracy results from our app cannot be extrapolated to any true results.

Reliability analysis was completed by calculating the Pearson correlation between the app and Push band for average concentric velocity and average peak velocity. With 5 participants each performing 20 total repetitions, there were a total of 100 data points to compare. However, due to issues with both the Push band and app incorrectly counting reps, the total data size dropped to 64 comparisons.

We calculated a Pearson correlation of 0.6 out of 1 for average concentric velocity and a Pearson correlation of 0.42 out of 1 for peak concentric velocity. According to work by Martinez-Cava et al., the benchmark for acceptable reliability is as high as > 0.990 [18]. Clearly, this result is not accurate enough for our app to be of use for VBT practice.

Interestingly, the first three participants showed a Pearson correlation of 0.94 out of 1 for average concentric velocity. These first three participants included the two novice lifters and one advanced lifter. It was only with the inclusion of the two intermediate-advanced lifters where the correlation dropped drastically. Certainly, the first three lifters could've just so happened to share similar velocity outputs between the app and Push band. After all, it is quite a small sample size. Theoretically though, this could be due to the fact that the intermediate-advanced lifters could achieve much faster velocities. It could be reasonably expected that these faster velocities caused not only some issues of reliability within our app but also within the Push Band device. Moreover, the two intermediate-advanced lifters were much more confident at each weighted increase, resulting in a more relaxed, less strict execution of their sets. This included things like pausing half-way down the eccentric portion, bouncing out of the bottom of the squat, and swaying back and forth before beginning their set. It was these intricacies that caused problems for both the app and the Push band. Nonetheless, studies measuring reliability of VBT devices typically employ multiple devices to test and to compare against, not to mention at least 15 participants per study across multiple days of evaluation. For a truer accuracy evaluation we recommend employing the use of a LPT instead of a Push band, increasing the number of participants drastically, and enforcing controlled execution of the exercise.

6.2 Project Utility

As mentioned prior, app utility was lower on the priority list for this project. Even so, each participant in the accuracy study was also asked to reflect on their experience with the app. One woman noted the possibility of auto-regulating her workouts, especially around fatigue caused by the menstrual cycle. This highlights the potential for increasing accessibility in strength training as VBT better aligns workouts with the dynamic nature of the human body. Another participant, a college strength and conditioning coach, liked the ability to view each repetition in isolation but would've liked power output features. Based on his feedback, bar path color labels were added to separate the eccentric and concentric phase of each repetition. Lastly, another participant who considered themselves an experienced lifter, disliked the strictness of the tracking. Performing "unnecessary" movements such as bouncing, pausing, and/or swaying, confused the algorithm and caused us to restart the evaluation. While this makes sense as an issue, it becomes difficult to incorporate this kind of robustness while maintaining automatic rep detection and barbell tracking. To address this, we added a manual way to begin and end velocity tracking so as to allow for looser unracking/racking. Other recommendations and insights will hopefully be applied in the future.

To note is that these discussions miss multiple key determinants of project utility. A longitudinal study might reveal issues of adherence, day-to-day bugs, or potential future features. No testing was done in different gyms, with different phone set ups, or with lifter's who had previous experience with VBT. Not to mention, it is unclear how this app might fit in the regular workflow of a lifter who normally doesn't employ VBT practices. Unfortunately, a study over a larger period of time is out of scope for this project.

6.3 Project Conclusion

Unquestionably, we did not reach our mark in terms of pure velocity tracking accuracy. Yet, as a proof of concept, the app delivered in terms of cost and potential utility. Many unexplored settings within the framework created by this app still remain to be explored, many of which could certainly improve the accuracy and usability of the app. For instance, changing AruCo tag parameters might improve tag detection speed and accuracy. Possibly, some combination of a colored contour with additional inner codification could improve motion blur resistance while still maintaining decent error correction. Considering other features, allowing for more manual rep counting could prevent erroneous movement from biasing the velocity statistics. And lastly, a more fleshed out user interface could certainly provide better usefulness for experienced practitioners and more unguided accessibility to beginners. All in all, while velocity accuracy did not meet the mark, there remains many paths to develop further, so in that sense the project proved the feasibility of a video-based barbell velocity and path tracker.

7 Ethical Considerations

As a concluding section, there exists various ethical considerations to be weary of, especially with the resources currently available for this project. These issues are directly linked to the main goals of the project. That is, providing barbell tracking accurately, increasing accessibility by lowering associated costs, and allowing tracking to occur over time. Primary among these considerations are the potential of providing faulty tracking to users and privacy issues concerning maintaining user video and velocity data for future review. Individually, these potential issues can be addressed, but as a whole, they might entail spending too much time on disparate features. Moreso, the current development plan of the project involves building solely the velocity and path tracking model before implementing it into a user application. This means that ethical concerns with app design will not be a main concern in this beginning stage, which could cause problems in the future. As such, we spell out the two primary concerns below.

7.1 Providing Accurate Velocity

Paramount to the utility of the project is the need for accurate tracking of bar path and bar velocity. Inherent with VBT methods is the fact that slight variances in velocity indicate important information about neuromusclar and functional performance [6]. It is these changes that velocity tracking hopes to analyze. Thus, one of the largest potential issues with the project is that large enough deviations from true velocity scores have the potential to invalidate the training method.

While deviations on the lower end might only lead to under-prescribing lifting intensity, deviations on the higher end have the potential to injure lifters relying on the application. When speaking on the ethical considerations of AI-based health-related apps, researcher Michael Kühler argues that "AI health apps... arguably lead to the novel issue of AI paternalism, particularly in the health care domain". That is, health and fitness apps are an instance of "persuasive technology" whereby the app relies on strategies (gamification, positive feedback, etc.) to motivate users towards specific ends [16]. The largest criticism of paternalism is the idea that it relies on an external notion of what is good for a person. A main advantage of VBT over traditional percentage-based training is the added personalizability that comes with basing training on daily and historic metrics. Nonetheless, this added personalizability gives VBT the allure of objectivity when it comes to prescribing exercise volume and load, meaning dedicated lifters or even novices might force themselves to perform more than they can handle. This can lead to problems of injury, overtraining, or burn-out, all issues that are contrary to the stated goal of the app.

The fact of the matter is that these issues do not exist if the app can provide sufficient accuracy as well as present the information in a way that maintains a user's autonomy. However, the idea of providing necessary accuracy is much easier said than done. Currently, there exists no mobile applications on the market that have been fully validated by existing scientific literature: ”While there is conflicting evidence, it appears that substantial bias and error can be introduced when different devices and/or users implement these measuring tools” [28]. Therefore, it is unlikely that our project will provide accurate enough velocity metrics to be reliably used for comprehensive velocity-based programming. In total, this means we must be careful about the framing of the project and its velocity output and recommendations.

7.2 Privacy of Video Data

Certainly, seeing bar path and velocity metrics on a given day allows lifters and coaches to modify and improve their daily resistance training programming. Even more useful is tracking those metrics across time, as a lifter/coach gains an understanding of a lifter's progression, performance compared to previous sessions, and patterns in training. So, our project aims to deliver a feature based on providing and comparing metrics across time. However, this comes with its own issues and it now requires the application to store user data. This might include videos of exercises being performed, velocity metrics per day, or simplified bar path graphs. As a result, there are new ethical considerations in terms of privacy concerns.

While exercise video data is not a typical form of health data, it still must be kept privately. As Kühler points out ”even if AI health apps' paternalistic influence on users were beneficial for them and did not undermine or diminish their autonomy, such apps might still end up being considered ethically problematic” [16]. Barbell tracking would stand to improve by tapping into user data. As our application would store the videos and metrics of our users, there is a potential for a substantial repository of data to pull from. Yet, this is exactly the issue with storing data, specifically, health and fitness data. As an application, users need not just be people interested in weight lifting. Perhaps, a user might be a physical therapist using the application to help their patients improve their movement. This would truly mean the application is storing health data. As a result, using user data will require informed consent and opacity on our end as developers. Additionally, using our application in a gym setting might inadvertently cause non-participating people to be recorded within the video. This is especially problematic because those recordings would be without the person's knowledge and there would be virtually no way to gain that informed consent. Some gyms even forbid video recording outright without prior permission from the staff. As such, our application sits in a difficult spot between providing high-granularity features and maintaining user and public privacy.

Less outwardly questionable is the issue of simply keeping the data private from external sources. To mitigate this potential problem, Schaar recommends employing, *Privacy by Design* including minimizing data collection, data sovereignty, and right to information among a handful of other concerns [23]. There does not seem to appear any potential features of the application that would deny these recommendations, but it will lead to longer development time. Questions we should be asking include: Does our application need a password feature? Should we store the entire video or just a reduced bar path representation? Should users be able to export and share their videos with others? These questions are somewhat difficult to address considering the projected development of the application starts not with front-end design but by focusing on creating highly accurate bar path and velocity tracking features. Regardless, storing health data of app users necessitates careful consideration in terms of data usage and data privacy.

References

[1] Balsalobre-Fernández, C and Torres-Ronda, L. “The Implementation of Velocity-Based Training Paradigm for Team Sports: Framework, Technologies, Practical Recommendations and Challenges”. In: *Sports (Basel)* (2021).

[2] Balsalobre-Fernández, C et al. “Analysis of Wearable and Smartphone-Based Technologies for the Measurement of Barbell Velocity in Different Resistance Training Exercises”. In: *Frontiers in Physiology* (2017).

[3] Balsalobre-Fernández, C et al. “Validity and reliability of a novel iPhone app for the measurement of barbell velocity and 1RM on the bench-press exercise”. In: *Journal of Sports Sciences* (2018).

[4] Boris, B, Yang, M, and Belongie, S. "Visual Tracking with Online Multiple Instance Learning". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[5] Courel-Ibáñez, J. et al. "Technical Note on the Reliability of the PowerLift App for Velocity-Based Resistance Training Purposes: Response". In: *Annals of Biomedical Engineering* (2020).

[6] Dorell, H.F, Smith, M.F, and Gee, T.I. "Comparison of Velocity-Based and Traditional Percentage-Based Loading Methods on Maximal Strength and Power Adaptations". In: *J Strength Cond Res.* (2020). URL: `https://pubmed.ncbi.nlm.nih.gov/30946276/`.

[7] Fiala, M. "ARTag, a fiducial marker system using digital techniques". In: *Computer Vision and Pattern Recognition* (2005).

[8] Garrido-Jurado, S. et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* (2014). URL: `https://doi.org/10.1016/j.patcog.2014.01.005.`.

[9] Grabner, H, Grabner, M, and Bischof, H. "Real-Time Tracking via On-line Boosting". In: *Proceedings of British Machine Vision Conference (BMVC)* (2006).

[10] He, K et al. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[11] Henriques, J et al. "High-Speed Tracking with Kernelized Correlation Filters". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015).

[12] Howard, A et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *ArXiv* (2017).

[13] Janicijevic, Danica et al. "Comparison of the two most commonly used gold-standard velocity monitoring devices (GymAware and T-Force) to assess lifting velocity during the free-weight barbell back squat exercise". In: *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology* 0.0 (2021), p. 17543371211029614. DOI: `10.1177/17543371211029614`. URL: `https://doi.org/10.1177/17543371211029614`.

[14] Kasovic, J et al. "Agreement Between the Iron Path App and a Linear Position Transducer for Measuring Average Concentric Velocity and Range of Motion of Barbell Exercises". In: *Journal of Strength and Conditioning Research* (2021).

[15] Kato, H. and Billinghurst, M. "Marker tracking and hmd calibration for a video-based augmented reality conferencing system". In: *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on. IEEE* (1999).

[16] Kühler, M. "Exploring the phenomenon and ethical issues of AI paternalism in health apps". In: *Bioethics* (2022). URL: `https://doi.org/10.1111/bioe.12886`.

[17] Liu, W et al. "SSD: Single Shot MultiBox Detector". In: *Computing Research Repository (CoRR)* (2015).

[18] Martínez-Cava, A et al. "Reliability of technologies to measure the barbell velocity: Implications for monitoring resistance training". In: *PLOS One* (2020).

[19] Pérez Castilla, A et al. "Reliability and Concurrent Validity of Seven Commercially Available Devices for the Assessment of Movement Velocity at Different Intensities During the Bench Press". In: *The Journal of Strength and Conditioning Research* (2019).

[20] Pueo, B et al. "Video-Based System for Automatic Measurement of Barbell Velocity in Back Squat". In: *Sensors (Basel)* (2021).

[21] Redmon, J et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* (2015).

[22] Ren, S et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *CoRR* abs/1506.01497 (2015).

[23] Schaar, P. "Privacy by Design". In: (2010). URL: `https://doi.org/10.1007/s12394-010-0055-x`.

[24] Simonyan, K and Zisserman, A. "Very Deep Convolutional Networks For Large-Scale Image Recognition". In: (2015).

[25] Tomasevicz, CL. et al. "Validity and reliability assessment of 3-D camera-based capture barbell velocity tracking device." In: *J Sci Med Sport* (2020). URL: `https://doi.org/10.1016/j.jsams.2019.07.014`.

[26] Wadhi, T et al. "Validity and Reliability of the GymAware Linear Position Transducer for Squat Jump and Counter-Movement Jump Height". In: *Sports (Basel)* (2018).

[27] Wang, J. and Olson, E. *AprilTag 2: Efficient and robust fiducial detection.* 2016.

[28] Weakley, J et al. "Velocity-Based Training: From Theory to Application". In: *Strength and Conditioning Journal* (2021).